

## THE PAN-STARRS DATA PROCESSING SYSTEM

EUGENE A. MAGNIER,<sup>1</sup> K. C. CHAMBERS,<sup>1</sup> H. A. FLEWELLING,<sup>1</sup> J. C. HOBLITT,<sup>2</sup> M. E. HUBER,<sup>1</sup> P. A. PRICE,<sup>3</sup> W. E. SWEENEY,<sup>1</sup> C. Z. WATERS,<sup>1</sup> L. DENNEAU,<sup>1</sup> P. DRAPER,<sup>4</sup> K. W. HODAPP,<sup>1</sup> R. JEDICKE,<sup>1</sup> N. KAISER,<sup>1</sup> R.-P. KUDRITZKI,<sup>1</sup> N. METCALFE,<sup>4</sup> C. W. STUBBS,<sup>5</sup> R. J. WAINSCOT<sup>1</sup>

*Draft version December 20, 2016*

### ABSTRACT

The Pan-STARRS Image Processing Pipeline performs the processing needed to download, archive, and process all images obtained by the Pan-STARRS telescopes. This article describes the overall data analysis system.

*Subject headings:* Surveys:Pan-STARRS 1

### 1. INTRODUCTION

This is the second in a series of seven papers describing the Pan-STARRS1 Surveys, the data reduction techniques and the resulting data products. This paper (Paper II) describes how the various data processing stages are organised and implemented in the Imaging Processing Pipeline (IPP), including details of the the processing database which is a critical element in the IPP infrastructure.

Chambers et al. (2017, Paper I) provides an overview of the Pan-STARRS System, the design and execution of the Surveys, the resulting image and catalog data products, a discussion of the overall data quality and basic characteristics, and a brief summary of important results.

Waters et al. (2016, Paper III) describes the details of the pixel processing algorithms, including detrending, warping, and adding (to create stacked images) and subtracting (to create difference images) and resulting image products and their properties.

Magnier et al. (2016a, Paper IV) describes the details of the source detection and photometry, including point-spread-function and extended source fitting models, and the techniques for “forced” photometry measurements.

Magnier et al. (2016b, Paper V) describes the final calibration process, and the resulting photometric and astrometric quality.

Flewelling et al. (2016, Paper VI) describes the details of the resulting catalog data and its organization in the Pan-STARRS database. Huber et al. (2017, Paper VII) describes the Medium Deep Survey in detail, including the unique issues and data products specific to that survey. The Medium Deep Survey is not part of Data Release 1. (DR1)

The Pan-STARRS1 filters and photometric system have already been described in detail in Tonry et al. (2012).

*Note: These papers are being placed on arXiv.org to provide crucial support information at the time of the*

*public release of Data Release 1 (DR1). We expect the arXiv versions to be updated prior to submission to the Astrophysical Journal in January 2017. Feedback and suggestions for additional information from early users of the data products are welcome during the submission and refereeing process.*

### 2. IPP SOFTWARE SUBSYSTEMS

#### 2.1. Processing Database

A critical element in the Pan-STARRS IPP infrastructure is the processing database. This database, using the mysql database engine, tracks information about each of the processing stages. It is used as the point of mediation of all IPP operations. Processing stages within the IPP perform queries of the database to identify the data to be processed at a given stage. As the processing for a particular stage is completed, summary information about the stage is written back to the database. In this way, the database records this history of the processing, and also provides the information needed to successive processing stages to begin their own tasks.

The processing database is colloquially referred to as the ‘gpc1’ database, since a single instance of the database is used to track the processing of images and data products related to the PS1 GPC1 camera. This same database engine also has instances for other cameras processed by the IPP, e.g., GPC2, the test cameras TC1, TC3, the Imaging Sky Probe (ISP), etc.

Within the processing database, the various processing stages are represented as a set of tables. In general, there is a top level table which defines the conceptual list of processing items either to be done, in progress, or completed. An associated table lists the details of elements which have been processed. For example, one critical stage is the Chip processing stage, discussed below, in which the individual chips from an exposure are detrended and sources are detected. Within the gpc1 database, there is a top-level table called ‘chipRun’ in which each exposure has a single entry. Associated with this table is the ‘chipProcessedImfile’ table, which contains one row for each of the (up to 60) chips associated with the exposure. The top-level tables, such as chipRun, are populated once the system has decided that a specific item (e.g., an exposure) should be processed at that stage. Initially, the entry is given a state of ‘run’, denoting that the exposure is ready to be processed. The low-level table entries, such as the chipProcessedImfile

<sup>1</sup> Institute for Astronomy, University of Hawaii, 2680 Woodlawn Drive, Honolulu HI 96822

<sup>2</sup> LSST Project Management Office, Tucson, AZ, U.S.A

<sup>3</sup> Department of Astrophysical Sciences, Princeton University, Princeton, NJ 08544, USA

<sup>4</sup> Department of Physics, Durham University, South Road, Durham DH1 3LE, UK

<sup>5</sup> Harvard-Smithsonian Center for Astrophysics, 60 Garden Street, Cambridge, MA 02138

entries, are only populated once the element (e.g., the chip) has been processed by the analysis system. Once all elements for a given stage, e.g., chips in this case, are completed, then the status of the top-level table entry (chipRun) are switched from ‘run’ to ‘done’.

If the analysis of an element (e.g., chip) completed successfully, then the corresponding table row (e.g., chipProcessedImfile) is listed with a fault of 0. If the analysis failed, then a non-zero fault is recorded. An analysis which results in a fault is one in which the failure is thought to be temporary. For example, if a computer had a network glitch and was unable to write out some of the output files, this would be an ephemeral failure which was not a failing of the data, but merely the processing system. On the other hand, if the analysis failed because of a problem with the input data, this is noted by setting a non-zero value in a different table field, ‘quality’. For example, if the chip analysis failed to discover any stars because the image was completely saturated, the analysis can complete successfully (fault = 0), but the ‘quality’ field will be set to a non-zero value. The various processing stages are able to select only the good (quality = 0) elements of a prior stage when choosing items for processing. For example, the Camera calibration stage will only use data from chips with good quality data, dropping the failed chips from the rest of the analysis. On the other hand, a fault in one of the elements for a given stage will block any dependent stages from processing that item. In this way, if a glitch occurs and a chip from an exposure failed to be written to disk in the Chip stage, the system will not partially process the exposure with the rest of the chips. Since many of the faults which occur are ephemeral, the processing stages are set up to occasional clear and re-try the faulted entries. Thus, automatic processing is able to keep the data flowing even in the face of occasional network glitches or hardware crashes.

## 2.2. Nebulous

Storing the large volume of data that is generated by the GPC1 camera was recognized early in the Pan-STARRS project as a major concern. The *Nebulous* system was designed to organize this data. The main components of this system is a database storing the locations of the files, with a Simple Object Access Protocol interface between the database and the other IPP programs. The actual files are stored on NFS mounted partitions on a series of storage nodes in the IPP cluster.

The original design of *Nebulous* intended to aid in the targetted processing of data, by having specific image data (such as all the images from one OTA device) located on a single storage node. This would allow any jobs involving that data to be assigned to the storage node, eliminating network IO. Important data could be duplicated to a further data nodes, with the alternate locations stored in the database. In practice, however, hardware failures and increases in hard drive storage volumes and network bandwidth have reduced the amount that the IPP processing is targeted.

When a program creates a new file in *Nebulous*, it supplies an URI of the form `neb://HOST.VOLUME/PATH/FILENAME`. The host and volume specifiers are optional, and allow a file to be created on a specific node. The path and filename

appear as a standard full file location, and are used internally as the “external id”. A `storage_object` is then created in the database for this id, and an instance of the file created on the specified node (or at random from available nodes if left empty).

## 2.3. Pantasks & Parallel Processing

### 2.4. DVO

The Pan-STARRS IPP uses an internal database system, distinct from the publically visible database system, to determine the association between multiple detections of the same astronomical object and as part of the astrometric and photometric calibration process. This database system, called the “Desktop Virtual Observatory” (DVO) was developed originally for the LO-NEOS project, and used as part of the CFHT Elixir system (Magnier & Cuillandre REF). The capabilities of this databasing system have been somewhat expanded for the Pan-STARRS context.

One of the main purposes of the DVO system is to define the relationship between individual detections of an astronomical object and the definition of that object. Before describing the database schema, we will discuss the process by which detections are associated with objects. New detections are generally added to the database in a group associated with, for example, an image from the GPC1 camera. As new detections are loaded, they are compared to the objects already stored in the database. If an object is already found in the database within the match radius, the new detection is associated to that object. If more than one object exists within the database, the detection is associated with the closest object.

Detections in DVO have a special piece of metadata called the **photcode** which identifies the source of the measurement. A **photcode** has a name which in general consists of the name of the camera or telescope (e.g., GPC1 or 2MASS), the name (or short-hand name) of the filter used for the measurement (e.g., *g*), and an identifier for the detector, if not unique (e.g., XY01 for GPC1). Along with each name, there is a numerical value for the photcode. A table within the DVO system, **Photcode**, lists the photcodes and defines a number of additional pieces of information for each photcode. These include the nominal zero point and airmass slope, as well as color trends to transform a measurement in the specific photcode to a common system. There are 3 classes of photcodes defined within the DVO system. Those photcodes associated with detections from an image loaded into the database system are called DEP photcodes. There are also photcodes associated with the average photometry values, called SEC photcodes. There are also those measurements which come from external data sources for which DVO does not have any information to determine a calibration (e.g., instrumental magnitudes and detector coordinates). These are measurements are reference values and are assigned REF photcodes.

In the implementation of DVO used for the PV3 calibration analysis, the database tables are stored on disk using binary FITS tables. Each type of database table is stored as a separate file, or a collection of files for table which are spatially partitioned. The binary FITS tables may be optionally compressed using the (to date) experimental FITS binary table compression strategy outlined

by REF. In this compression scheme, using a strategy similar to that used for FITS image compression (REF), the data stored in the binary table is compressed and stored in the 'HEAP' section of the FITS table. In brief, each column in the FITS table is compressed as one (or more) blocks. The standard fields which describe the data column format (e.g., TFORM1) are replaced with columns which describe the location and size of the compressed data in the HEAP section; the information about the uncompressed data is moved to a field with 'Z' prepended (e.g., ZFORM1) and an additional field is added to define the compression algorithm (e.g., ZCTYP1). The column names (e.g., TTYPE1) and units (e.g., TUNIT1) are retained in their original form. The compression algorithm can treat the entire column as a single block of data, or it may be broken into a number of chunks, each compressed in turn (this must be the same for all columns). Additional header information is added to describe the block sizes and information needed to describe the HEAP data section. The compression algorithms currently defined consist of the GZIP, RICE, PLIO, and HCOMPRESS (REFS). For GZIP, the compression algorithm may transpose the byte order before compression: for floating point data of a similar dynamic range, this choice may allow for better compression as each byte in the 4 or 8 byte floating point value is more likely to be similar to the same byte in other rows than to the other bytes of the same row value. This option is called GZIP\_2 in the FITS standard, as opposed to the standard order, GZIP\_1. The DVO system can be set to specify the compression options for each column in the tables. In practice, we have chosen a default in which floating point numbers used GZIP\_2, character strings use GZIP\_1, integers use RICE.

#### 2.4.1. Sky Partition

DVO includes two major classes of database tables: those containing information directly about astronomical objects in the sky and those containing other supporting information. The object-related tables are partitioned on the basis of position in the sky: objects within a region bounded by lines of constant RA, DEC are contained in a specific file. The boundaries and the associated partition names are stored in one of the supporting tables, **SkyTable**. This table contains the definitions of the boundaries for each sky region (R\_MIN, R\_MAX, D\_MIN, D\_MAX), the name of the sky region, an ID (INDEX, equal to the sequence number of the region in the table), and index entries to enable navigation within the table. The regions are defined in a hierarchical sense, with a series of levels each containing a finer mesh of regions covering the sky.

In the default used by the PV3 DVO, the partitioning scheme is based on the one used by the Hubble Space Telescope Guide Star Catalog files. Level 0 is a single region covering the full sky. Level 1 divides the sky in Declination into bands  $7.5^\circ$  high. Level 2 subdivides these Declination bands in the RA direction, with spacing related to the stellar density. Level 3 divides these RA chunks into 4 - 8 smaller partitions. This level exactly matches the HST GSC layout, and uses the same naming convention to identify the partitions: n0000/0000, etc. **more on the names?** Level 4 further divides these regions by a factor of 16. In the **SkyTable**, a region at one

level has a pointer to its parent region (the one which contains it) and a sequence pointing to its children (regions it contains). The **SkyTable** enables fast lookups of the on-disk partitions which map to a specific coordinate on the sky. In general, a single DVO will have the full sky represented with tables at a single level, though it is possible for mixed levels to be used, this mode is not well tested. For the PV3 master database, the partitioning at the 5th level results in  $\sim 150,000$  regions to cover the full sky, of which  $\sim 110,000$  are used for the PV3  $3\pi$  data. The densest portions of the bulge contain at most  $\sim 300k$  astronomical objects in the database files, with an associated maximum of 30M measurements in these files. With the compression scheme described above, this makes the largest database files  $\sim 3GB$ , which can be loaded into memory in 30 seconds on our partition machines.

The DVO software system allows the tables which are partitioned across the sky to also be distributed across multiple computers, which we call partition hosts. A single file defines the names of these partition hosts and the location of the database partition on the disks of that machine. The **SkyTable** contains elements to define by ID the partition host to which a partitioned set of tables has been assigned. Operations which query the database, or perform other operations on the database, are aware of the partitioning scheme and will launch their operations as remote processes on the machines which contain the data they need. For example, a query for data from a small region will launch sub-query operations on the machines which contain the data overlapping the region of interest. These remote query operations will select the database information which matches the query request (i.e., applying restrictions as defined) and return to the master process the results. The results from the various partition hosts are then merged into a single result by the master process. This parallelization is critical to querying and manipulating the enormous database on a reasonable timescale.

#### 2.5. Tables which describe objects

Two tables carry the most important information about the astronomical objects in the database: Average and SecFilt. These two tables specify the main average properties of the astronomical object. The Average table includes the astrometric information ( $\alpha, \delta, \mu\alpha, \mu\delta, \pi$ ) and associated errors, data quality flags for each object, links to the other tables, and a number of IDs, with one row for each astronomical object. **go into complete detail here on the IDs?** The SecFilt table<sup>6</sup> contains average photometric information for a collection of filters. A given DVO instance has a specified set of filters for which average photometry is stored in the SecFilt table. The number and choice of filters for the SecFilt table can be modified by the database administrator fairly easily, but the process of updating the database is somewhat expensive ( $\sim 24$  hours for the current PV3 instance). Thus

<sup>6</sup> The name SecFilt is a bit of a historical misnomer: originally, DVO was designed for a monochromatic survey and data for a single photometric band was maintained in the Average table. Later, DVO was adapted to a multifilter system and additional filters were added to the SecFilt (Secondary Filter) table. Eventually, the schema was normalized and all photometric data placed in SecFilt, with the Primary filter concept being dropped, but the name has since stuck.



the choice is semi-static for a given DVO instance. In the case of the PV3 DVO instance, 9 average bandpasses are defined:  $g, r, i, z, y, J, H, K, w$ . The SecFilt table contains one row for each filter for each object, thus the PV3 DVO contains 9 times as many rows as the Average table. The order of the table is fixed in relation to the Average table: row  $i$  of Average defines the object with photometry contained in rows  $9i \rightarrow 9i + 8$  ( $i$  is zero counting).

The individual measurements of the astronomical objects are carried in the table **Measure**. This table lists the values measured by **psphot** for each chip, warp, or stack image. This includes the instrumental magnitudes for the PSF, aperture, and Kron photometry; raw position ( $X_{ccd}, Y_{ccd}$ ) and second moments ( $M_{xx}, M_{yy}, M_{xy}$ ), along with shape parameters of the PSF model at the position of the object ( $FWx, FWy, \theta$ ). This table also includes metadata such as the exposure time, the date & time of the observation, airmass, azimuth, and information specifying the filter **describe the phot-codes**. The **Measure** table also carried the calibration magnitude offsets ( $M_{cal}$  and  $M_{flat}$  discussed below) and the astrometrically calibrated position. Astrometric offsets for several systematic corrections discussed below are also defined for each measurement. Since stacks and forced warp photometry may have non-significant values, the table is somewhat de-normalized in that it also carried instrumental flux values for the PSF, aperture, and Kron photometry.

In the **Measure** table, there are three fields which provide two independent links from the specific measurement to the associated object: **Measure.catID** specifies the spatial partition to which the measurement belongs; **Measure.objID** specifies to which entry in the **Average** table the measurement belongs. These two 32 bit fields can thus be combined into a single 64 bit unique ID for all objects in the database. In addition, the field **Measure.averef** specifies the row number in the **Average** table of the associated object. The **Measure** table may be unsorted, in which case it is slow to find the measurements associated with a specific object (a full table scan is required). After the table is sorted, the **Measure** rows for a given object are grouped together. In the case, the fields **Average.measureOffset** and **Average.Nmeasure** define an index for the code to jump to the list of measurements for a single object. The field **Measure.imageID** defines the link from the measurement to the image which supplied the measurement.

**some discussion of the db construction, addstar, dvomerge, etc?**

For the warp images, we also measure the weak lensing KSB parameters related to the shear and smear tensors. These measurements are stored in the **Lensing** table, along with the radial aperture fluxes for radii numbers 5, 6, & 7 (XX, XX, XX arcsec). This table contains one row for every warp row. Similarly to the **Measure** table, the fields **objID**, **catID**, and **averef** define links from the **Lensing** table to the **Average** table. In a similar fashion, the fields **Average.lensingOffset** and **Average.Nlensing** are the index into the sorted **Lensing** table entries. **discuss failure of the Lensing to Measure indexing**

The values stored in the **Lensing** table are used to calculate average values for each of these types of mea-

surements in each filter. The **Lensobj** table stores the averaged KSB and radial aperture photometry for each of the 5 filters  $g_{P1}, r_{P1}, i_{P1}, z_{P1}, y_{P1}$ . This table contains one entry per object per filter. The table is not generally stored unsorted as it is calculated after the full database is populated. The link from **Average** to **Lensobj** is defined by the fields **Average.offsetLensobj** and **Average.Nlensobj**. Each **Lensobj** row also includes the photcode (filter) for which the average lensing (and radial aperture) properties have been calculated.

The **Galphot** table stores the results of the forced galaxy fitting analysis for each object that has been measured. This table contains one row per filter and model type (Sersic, Exponential, DeVaucouleur) if forced galaxy models have been calculate for the object. **need to expand on this somewhat**

The **Starpar** table carries measurements provide by Greg Green & Eddie Schlafly from their analysis of the SED of objects in the PS1  $3\pi$  data, using the **PV1?** version of the analysis (Green et al REF). In this work, the goal was a 3D model of the dust in the Galaxy based on Pan-STARRS **(and WISE & 2MASS?)** photometry. As part of this analysis, the authors fit the SEDs of all **stellar?** sources with stellar models including free parameters of extinction, distance modulus, metallicity, and absolute r-band magnitude. While these photometric distance modulus measurements are not extremely precise (see below), they provide a constraint on the distance is used in our analysis of the astrometry (see Section ??).

## 2.6. Other Tables

Data from GPC1 (and other cameras processed by the IPP) are loaded into DVO in units **smf** files generated by the Camera calibration stage. As described above, these files contain all measurements from a complete exposure, with measurements from each chip grouped into separate FITS tables. When these measurements are loaded into the **Measure** and similar tables, a subset of the information from the chip header is used to populated a row in the DVO **Image** table. This table contains one row for each chip known to DVO, with information such as the filter (**photcode**), the exposure time, the airmass, the astrometric calibration terms, the photometric zero point, etc. For GPC1 and other mosaic cameras, an additional row is defined to carry the projection and camera distortion elements of the astrometry model. As chips are loaded into this table, they are assigned an internal ID (a running sequence in the table). Images may also be assigned an external ID: in the case of the GPC1 images, this ID is defined by the processing mysql database and is guaranteed to be unique within the processing system.

Other tables are used to track information used by the calibration system. This includes the complete set of flat-field corrections determined by the photometry calibration analysis (see Section ??) and the astrometric flat-field corrections determined by the astrometry calibration analysis (see Section ??)

## 3. IPP DATA PROCESSING STAGES

### 3.1. Download from Summit

As exposures are taken by the PS1 telescope & camera system, the 60 OTA CCDs are read out by the camera

software system and each chip is written to disk on computers at the summit in the PS1 facility. The chip images are written to a collection of machines in the PS1 facility called the ‘pixel servers’. After the images are written to disk, a summary listing of the information about the exposure and the chip images are written to an http server system called the ‘datastore’. The datastore exposes, via http, a list of the exposures obtained since the start of the PS1 operations. Requests to this server may restrict to the latest by time. Each row in the listing includes basic information about the exposure: an exposure identifier (e.g., o5432g0123o; see ?? for details), the date and time of the exposure, *etc.* The row also provides a link to a listing of the chips associated with that exposure. This listing includes a link to the individual chip FITS files as well as an md5 CHECKSUM. Systems which are allowed access may download chip FITS files via http requests to the provided links.

During night-time operations, while the telescope is observing the sky and the camera subsystem is saving images to the pixel servers and adding their information to the datastore list, the IPP subsystem called ‘summitcopy’ monitors the datastore in order to discover new exposures ready for download. Once a new exposure has been listed on the datastore, summitcopy adds an entry of the exposure to a table in the processing database (summitExp). This tells the summitcopy to look for the list of chips, which are then added to another table (summitImfile). The summitcopy system then attempts to download the chips from the summit pixel servers with an http request. As the chip files are downloaded, their md5 checksum values are calculated and compared with the value reported by the camera subsystem / datastore. Download failures are rare and marked as a non-zero fault, allowing for a manual recovery, rather than automatically rejecting the failed chips.

### 3.2. Image Registration

Once chips for an exposure have all been downloaded, the exposure is ready to be registered. In this context, ‘registration’ refers to the process of adding them to the database listing of known, raw exposures (not to be confused with ‘registration’ in the sense of pixel realignment). The result of the registration analysis is an entry for each exposure in the rawExp table, and one for each chip in the rawImfile table. These tables are critical for downstream processing to identify what exposures are available for processing in any other stage. In the registration stage, a large amount of descriptive metadata for each chip is added to the rawImfile table, some of which is extracted from the chip FITS file headers (e.g., RA, DEC, FILTER) and some of which is determined by a quick analysis of the pixels (e.g., mean pixel values, standard deviation). The chip-level information is merged into a set of exposure-level metadata and added to the rawExp table entry. The exposure-level metadata may be the same as any one of the chip, in a case where the values are duplicated across the chip files (e.g., the name of the telescope or the date & time of the exposure), or it may be a calculation based on the values from each chip (e.g., average of the average pixel values).

Unlike much of the rest of the IPP stage, the raw exposures may only have a single entry in the registration tables of the processing database tables (rawExp and raw-

Imfile).

### 3.3. Chip Processing

The science analysis of an exposure begins with the processing of the individual chips, the Chip Processing stage. This analysis step has two main goals: the removal of the instrumental signature from the pixel values (detrending) and the detection of the sources in the objects. In the Chip stage, the individual chips are processed independently in parallel within the data processing cluster. Within the processing computer cluster, most of the data storage resources are in the form of computers with large raids as well as substantial processing capability. The processing system attempts to locate one copy of specific raw chips on pre-defined computers for each chip. The processing system is aware of this data localization and attempts to target the processing of a particular chip to the machine on which the data for that chip is stored. The output products are then primarily saved back to the same machine. This ‘targetted’ processing was an early design choice to minimize the system wide network load during processing. In practice, as computer disks filled up at different rates, the data has not been localized to a very high degree. The targeted processing has probably reduced the network load somewhat but it has not been as critical of a requirement as originally expected.

The Chip processing stage consists of: reading the raw image into memory, applying the detrending steps (see Waters et al. 2016), stitching the individual OTA cells into a single chip image, detection and characterization of the sources in the image (see Magnier et al. 2016b), and output of the various data products. These include the detrended chip image, variance image, and mask image, as well as the FITS catalog of detected sources. The PSF model and background model are also saved, along with a processing log. A selection of summary metadata describing the processing results are saved and written to the processing database along with the completion status of the process. Finally, binned chip images are generated (on two scales, binned by 16 and 256 pixels) for use in the visualization system of the processing monitor tool.

### 3.4. Camera Calibration

After sources have been detected and measured for each of the chip, the next stage is to perform a basic calibration of the full exposure. This stage starts with the collection of FITS tables containing the instrumental measurements of the detected sources, primarily the positions ( $x_{ccd}$ ,  $y_{ccd}$ ) and the instrumental PSF magnitudes. The data for all chips of an exposure are loaded by the analysis program. The header information is used to determine the coordinates of the telescope boresite (RA, DEC, Position angle). These three coordinates are used, along with a model of the camera layout, to generate an initial guess for the astrometry of each chip. Reference star coordinates and magnitudes are loaded from a reference catalog for a region corresponding to the boundaries of the exposure, padded by a large fraction of the exposure diameter in case of a modest pointing error. The guess astrometry is used to match the reference catalog to the observed stellar positions in the focal plane coordinate system. Once an acceptable match is found, the astrometric calibration of the individual chips

is performed, including a fit to a single model for the distortion introduced by the camera optics. After the astrometric analysis is completed, the photometric calibration is determined using the final match to the reference catalog. At this stage, pre-determined color terms may be included to convert the reference photometry to an appropriate photometric system. For PS1, this is used to generate synthetic w-band photometry for areas where no PS1-based calibrated w-band photometry is available. For more details, see Magnier et al. (2017).

In addition to the astrometric and photometric calibrations, the Camera stage also generates the dynamic masks for the images. The dynamic masks include masking for optical ghosts, glints, saturated stars, diffraction spikes, and electronic crosstalk. The mask images generated by the Chip stage are updated with these dynamic masks and a new set of files are saved for the downstream analysis stages.

The Camera stage also merges the binned chip images (see ??) into single jpeg images of the entire focal plane. These jpeg images can then be displayed by the process monitoring system to visualize the data processing.

### 3.5. Warp

Once astrometric and photometric calibrations have been performed, images are geometrically transformed into a set of common pixel-grid images with simple projections from the sky. These images, called skycells, can then be used in subsequent stacking and difference image analysis without concern about the astrometric transformation of an exposure. This processing is called ‘warping’; the warp analysis stage is run on all exposures before they are processed further. For details on the warping algorithm, see Waters et al. (2016).

The output products from the Warp stage consist of the skycell images containing the signal, the variance, and the mask information. These images have been shipped to STScI and **are available / will be available** from the image extraction tools **in DR2**.

### 3.6. Stack

The skycell images generated by the Warp process are added together to make deeper, higher signal-to-noise images in the Stack stage. The stacks also fill in coverage gaps between different exposures, resulting in an image of the sky with more uniform coverage than a single exposure. See Waters et al. (2016) for details on the stack combination algorithm.

In the IPP processing, stacks may be made with various options for the input images. During nightly science processing, the 8 exposures per filter for each Medium Deep field are combined into a set of stacks for that field. These so-called ‘nightly stacks’ are used by the transient survey projects to detect the faint supernovae, among other transient events. For the PV3  $3\pi$  analysis, all filter images from the  $3\pi$  survey observation were stacked together to generate a single set of images with  $\sim 10 - 20\times$  the exposure of the individual survey exposures. The signal, variance, and mask images resulting from these deep stacks are part of the DR1 release and are available from the image extraction tools.

For the PV3 processing of the Medium Deep fields, stacks have been generated for the nightly groups and

for the full depth using all exposures (deep stacks). In addition, a ‘best seeing’ set of stack have been produced **using image quality cuts to be described**. We have also generated out-of-season stacks for the Medium Deep fields, in which all image not from a particular observing season for a field are combined into a stack. These later stacks are useful as deep templates when studying long-term transient events in the Medium Deep fields as they are not (or less) contaminated by the flux of the transients from a given season.

### 3.7. Stack Photometry

The stack images are generated in the Stack stage of the IPP, but the source detection and extraction analysis of those images is deferred until a separate stage, the Stack Photometry stage. This separation is maintained because the stack photometry analysis is performed on all 5 filter stack images at the same time. By deferring the analysis, the processing system may decouple the generation of the pixels from the source detection. This makes the sequencing of analysis somewhat easier and less subject to blocks due to a failure in the stacking analysis.

The stack photometry algorithms are described in detail in Magnier et al. (2016b). In short, sources are detected in all 5 filter images down to the  $5\sigma$  significance. The collection of detected sources is merged into a single master list. If a source is detected in at least two bands, or only in  $y$ -band, then a PSF model is fitted to the pixels of the other bands in which the source was not detected. This forced photometry results in lower significance measurements of the flux at the positions of objects which are thought to be real sources, by virtue of triggering a detection in at least two bands. The relaxed limit for  $y$ -band is included to allow for searches of  $y$ -dropout objects: it is known that faint, high-redshift quasars may be detected in  $y$ -band only. The casual user of the PV3 dataset should be wary of sources detected only in  $y$ -band as these are likely to have a higher false-positive rate than the other stack sources.

The stack photometry output files consist of a set of FITS tables in a single file, with one file for each filter. Within one of these files, the tables include: the measurements of sources based on the PSF model; aperture like parameters such as the Petrosian flux and radius; the convolved Galaxy model fits; the radial aperture measurements. **is this list complete?**

The stack photometry output catalogs are re-calibrated for both photometry and astrometry in a process very similar to the Camera calibration stage. In the case of the stack calibration, however, each skycell is processed independently. The same reference catalog is used for the Camera and Stack calibration stages.

### 3.8. Forced Warp Photometry

Traditionally, projects which use multiple exposures to increase the depth and sensitivity of the observations have generated something equivalent to the stack images produced by the IPP analysis. In theory, the photometry of the stack images produces the ‘best’ photometry catalog, with best sensitivity and the best data quality at all magnitudes (c.f, CFHT Legacy survey, COSMOS, etc). In practice, the stack images have some significant



limitations due to the difficulty of modelling the PSF variations. This difficulty is particularly severe for the Pan-STARRS  $3\pi$  survey stacks due to the combination of the substantial mask fraction of the individual exposures, the large intrinsic image quality variations within a single exposure, and the wide range of image quality conditions under which data were obtained and used to generate the  $3\pi$  PV3 stacks.

For any specific stack, the point spread function at a particular location is the result of the combination of the point spread functions for those individual exposures which went into the stack at that point. Because of the high mask fraction, the exposures which contributed to pixels at one location may be somewhat different just a few tens of pixels away. Because of the intrinsic variations in the PSF across an exposure and because of the variations from exposure to exposure, the distribution of point spread functions of the images used at one position may be quite different from those at a nearby location. In the end, the stack images have a effective point spread function which is not just variable, but changing significantly on small scales in a highly textured fashion.

Any measurement which relies on a good knowledge of the PSF at the location of an object either needs to determine the PSF variations present in the stack, or the measurement will be somewhat degraded. The highly textured PSF variations make this a very challenging problem: not would such a PSF model require an unusually fine-grained PSF model, there would likely not be enough PSF stars in an given stack to determine the model at the resolution required. The IPP photometry analysis code uses a PSF model with 2D variations using a grid of at most  $6 \times 6$  samples per skycell, a number reasonably well-matched to the density of stars at most moderate Galactic latitudes. This scale is far too large to track the fine-grained changes apparent in the stack images.

Thus PSF photometry as well as convolved Galaxy models in the stack are degraded by the PSF variations. Aperture-like measurements are in general not as affected by the PSF variations, as long as the aperture in question is large compared to the FWHM of the PSF.

The PV3  $3\pi$  analysis solves this problem by using the sources detected in the Stack images and performing forced photometry on the individual warp images used to generate the stack. This analysis is performed on all warps for a single filter as a single job, though this is more of a bookkeeping aid as it is not necessary for the analysis of the different warps to know about the results of the other warps.

In the forced warp photometry, the positions of sources are loaded from the stack outputs. PSF stars are pre-identified and a PSF model generated for each warp based on those stars, using the same stars for all warps to the extent possible (PSF stars which are excessively masked on a particular image are not used to model the PSF). The PSF model is fitted to all of the known source positions in the warp images. Aperture magnitudes, Kron magnitudes, and moments are also measured at this stage for each warp. Note that the flux measurement for a faint, but significant, source from the stack image may be at a low significance ( $< 5\sigma$ ) in any individual warp image; the flux may even be negative for specific warps. When combined together, these low-significance

measurements will result in a significant measurement as the signal-to-noise increases by  $\sqrt{N}$ .

### 3.9. Forced Galaxy Models

The convolved galaxy models are also re-measured on the warp images by the forced photometry analysis stage. In this analysis, the galaxy models determined by the stack photometry analysis are used to seed the analysis in the individual warps. The purpose of this analysis is the same as the forced PSF photometry: the PSF of the stack is poorly determined due to the masking and PSF variations in the inputs. Without a good PSF model, the PSF-convolved galaxy models are of limited accuracy.

In the forced galaxy model analysis, we assume that the galaxy position and position angle, along with the Sersic index if appropriate, have been sufficiently well determined in the stack analysis. In this case, the goal is to determine the best values for the major and minor axis of the elliptical contour and at the same time the best normalization corresponding to the best elliptical shape (and thus the best galaxy magnitude value).

For each warp image, the Stack value for the major and minor axis are used as the center of a  $7 \times 7$  grid search of the major and minor axis parameter values. The grid spacing is defined as a function of the signal-to-noise of the galaxy in the stack image so that bright galaxies are measured with a much finer grid spacing that faint galaxies **need to quantify this**. For each grid point, the major and minor axis values at that point are determined for the model. The model is then generated and convolved with the PSF model for the warp image at that point. The resulting model is then compared to the warp pixel data values and the best fit normalization value is defined. The normalization and the  $\chi^2$  value for each grid point is recorded.

For a given galaxy, the result is a collection of  $\chi^2$  values for each of the grid points spanning all warp images. A single  $\chi^2$  grid can then be made from all warps by combining each grid point across the warps. The combined  $\chi^2$  for a single grid point is simply the sum of all  $\chi^2$  values at that point. If, for a single warp image, the galaxy model is excessively masked, then that image will be dropped for all grid points for that galaxy. The reduced  $\chi^2$  values can be determined by tracking the total number of warp pixels used across all warps to generate the combined  $\chi^2$  values. From the combined grid of  $\chi^2$  values, the point in the grid with the minimum  $\chi^2$  is found. Quadratic interpolation is used to determine the major, minor axis values for the interpolated minimum  $\chi^2$  value. The errors on these two parameters is then found by determining the contour at which the **reduced?  $\chi^2$  increases by 1**.

Thus the Forced Galaxy Model analysis uses the PSF information from each warp to determine a best set of convolved galaxy models for each object in the stack images. **discuss the subset of galaxy models and objects**.

### 3.10. Difference Images

Two of the primary science drivers for the Pan-STARRS system are the search hazardous asteroids and the search for Type Ia supernovae to measure the history of the expansion of the universe. Both of these projects

require the discovery of faint, transient source in the images. For the hazardous asteroids, and solar system studies in general, the sources are transient because they are moving between observations; supernovae are stationary but transient in brightness. In both cases, the discovery of these sources can be enhanced by subtracting a static reference image from the image taken at a certain epoch. The quality of such a difference image can be enhanced by convolving one or both of the images so that the PSFs in the two images are matched. **discuss Alard-Lupton.**

In the Difference Image stage, the IPP generates difference images for specified pairs of images. It is possible for the difference image to be generated from a pair of warp images, from a warp and a stack of some variety, or from a pair of stacks. During the PS1 survey, pairs of exposures, call TTI pairs (see **Survey Strategy**), were obtained for each pointing within a  $\sim 1$  hour period in the same filter, and to the extent possible with the same orientation and boresite position. The standard PS1 nightly processing generated difference images from the resulting warp pairs ('warp-warp diffs').

The nightly stacks generated for the Medium Deep fields were combined with a template reference stack image to generate 'stack-stack diffs' for these fields each night.

For the PV3 processing, the entire collection of warps for the  $3\pi$  survey were combined with the  $3\pi$  stacks to generate 'warp-stack diffs'.

3.11. *Addstar : DVO Ingest*

3.12. *Calibration Operations*

3.13. *IPP to PSPS*

3.14. *PSPS Load & Merge*

4. IPP HARDWARE SYSTEMS

4.1. *Kihei Processing Cluster*

4.2. *Los Alamos National Labs*

In order to increase the processing rate for the 3II PV3 data, we partnered with Los Alamos National Lab to gain access to the Mustang supercomputer. The supercomputer is comprised of 3088 processing nodes, each with 12 cores and 64GB of RAM. The processing nodes do not have significant local disks, but are connected to multiple petabyte scale scratch disks. Job management is controlled by the Moab HPC system<sup>7</sup>, which schedules resource requests among users, allocating processing nodes to satisfy jobs, and terminating those jobs if they exceed their scheduled time limit.

This system is part of the lab's "Turquoise" network, allowing it to be used for research projects that do not handle sensitive data. It is, however, subject to stricter access controls than are in place at the main IPP processing cluster. Login sessions are terminated after 12 hours, requiring new sessions to be initiated regularly. Network access is also filtered, with only SSH connections allowed between the IPP cluster and Los Alamos. This restriction removes the ability for the processing to contact the IPP processing database directly.

To work around this, additional steps were needed to ensure efficient use of the computing resources. A periodic poll of outstanding tasks was done on the IPP clus-

**Table 1**  
Cost values for remote processing

IPP Stage	$t_{\text{task}}$ (s)	$S_{\text{task}}$
CHIP	150	2
CAMERA	1700	2
WARP	110	2
STACK	1500	6
STATISKY	7200	6
FULLFORCE	300	2

ter, using the information stored in the database, and those tasks assigned to a processing bundle. Each component task in the bundle was then checked to identify the set of input files needed to complete the task, the commands necessary to complete the task, and the set of output files that should be generated if the task completed successfully. Once this information had been generated for all tasks, the component lists were merged, and the Moab job control file was constructed.

The control file contains the resource requests for the job, as well as the commands to be executed to complete it. The resource request was calculated based on the number of tasks included in the job bundle  $N_{\text{tasks}}$ , and scaled by the expected execution time ( $t_{\text{task}}$ ) and computational intensity of the component tasks ( $S_{\text{task}}$ ). For a given job bundle, an initial estimate of the number of compute nodes needed is simply  $\text{nodes} = S_{\text{task}} * N_{\text{tasks}} / 12$ . To ensure that jobs were not prematurely terminated, we attempted to design the requested job processing time to be 25% longer than the expected time to complete the component tasks. Based on the initial node count, we calculated the request time as  $t_{\text{request}} = \lceil 1.25 \frac{\text{nodes} * t_{\text{task}}}{\text{nodes}_{\text{max}}} \rceil + 1$ , where  $\text{nodes}_{\text{max}}$  is the maximum number of nodes that can be requested in a single job (1000 for Mustang). Table 1 contains the cost values used for the various IPP processing stages.

Once the preparation for the job is complete, the input and output file lists, the task list, and the job control file are transferred via SCP to the Mustang cluster. Local tasks are then initiated on the user interface nodes to SCP the input files onto the scratch storage disks if they do not already exist. Once all the input files have been copied, the job is submitted to Moab for scheduling. The Moab interface is periodically polled to determine the job status, and after it has completed, the results are retrieved in a similar way. Local tasks again SCP the output products, but to copy the results back to the IPP cluster.

In addition to the standard output products, "dbinfo" files are constructed as part of the job execution. These files contain database update commands to ensure that the IPP processing database has the correct entries for the tasks that were remotely executed. These commands are executed after confirming that all retrieved output products are present.

Approximately half of the chip through warp processing for the PV3 reduction was performed on Mustang, with 201,040 / 375,573 of the CAMERA stage products reduced there. Only processing through the STACK stage was attempted, although with a smaller fraction of the total compared to the CAMERA stage, with 290,257 / 998,886 being produced at Los Alamos. One reason for this decrease is that due to the memory constraints on

<sup>7</sup> <http://www.adaptivecomputing.com/>



the Mustang processing nodes, we were unable to run stacks with more than 25 inputs there. Stacks with this larger number of inputs overflow the memory of the processing node, and as they do not have disk space available for use as virtual memory, cause the machine to hang until the job time limit is reached. These stacks were instead processed on the regular IPP cluster, where hosts with sufficient memory were available.

#### 4.3. UH Cray Cluster

In December 2014, the University of Hawaii installed a 178-compute node Cray supercomputer on the main Manoa campus. As part of the initial commissioning of this computer, Pan-STARRS was invited to use this resource in February 2015, roughly corresponding with the completion of the initial processing of the CHIP through STACK processing. Although the number of nodes was much smaller than that available on Mustang, the nodes were more robust, with 20 cores and 128 GB of memory. The scratch data storage was somewhat smaller than at Los Alamos, with only a single 600 TB volume. We had the unique ability to rapidly deploy to the UH Cray, using almost all nodes for IPP processing as other users at the university were designing code. This rapid deployment was made possible by the similarity of the Slurm<sup>8</sup> scheduler and tools to those used by Moab (although the UH Cray has a smaller nodes<sub>max</sub> of 10).

The UH Cray was used to do processing for the STATISKY stage, running approximately half of that photometry (101,528 / 200,720). We were also able to run part of the FULLFORCE photometry there as well, although more had to be run on the IPP cluster as other users started to utilize the system, with 168,685 / 994,890 runs processed there.

The Pan-STARRS1 Surveys (PS1) have been made possible through contributions of the Institute for Astronomy, the University of Hawaii, the Pan-STARRS Project Office, the Max-Planck Society and its participating institutes, the Max Planck Institute for Astronomy, Heidelberg and the Max Planck Institute for Extraterrestrial Physics, Garching, The Johns Hopkins University, Durham University, the University of Edinburgh, Queen's University Belfast, the Harvard-Smithsonian

Center for Astrophysics, the Las Cumbres Observatory Global Telescope Network Incorporated, the National Central University of Taiwan, the Space Telescope Science Institute, the National Aeronautics and Space Administration under Grant No. NNX08AR22G issued through the Planetary Science Division of the NASA Science Mission Directorate, the National Science Foundation under Grant No. AST-1238877, the University of Maryland, and Eotvos Lorand University (ELTE) and the Los Alamos National Laboratory.

#### REFERENCES

- Chambers, K. C., Magnier, E. A., Metcalfe, N., & et al. 2017, ArXiv e-prints
- <sup>8</sup> <https://slurm.schedmd.com/>
- Flewelling, H. A., Magnier, E. A., Chambers, K. C., Heasley, J. N., Holmberg, C., Huber, M. E., Sweeney, W., Waters, C. Z., Chen, T., Farrow, D., Hasinger, G., Henderson, R., Long, K. S., Metcalfe, N., Nieto-Santisteban, M. A., Norberg, P., Saglia, R. P., Szalay, A., Rest, A., Thakar, A. R., Tonry, J. L., Valenti, J., Werner, S., White, R., Denneau, L., Draper, P. W., Jedicke, R., Kudritzki, R.-P., Price, P. A., Chastel, S., McClean, B., Postman, M., & Shiao, B. 2016, ArXiv e-prints
- Huber, M., TBD, A., TBD, B., & et al. 2017, ArXiv e-prints
- Magnier, E. A., Schlafly, E. F., Finkbeiner, D. P., & et al. 2017, ArXiv e-prints
- Magnier, E. A., Schlafly, E. F., Finkbeiner, D. P., Tonry, J. L., Goldman, B., Röser, S., Schilbach, E., Chambers, K. C., Flewelling, H. A., Huber, M. E., Price, P. A., Sweeney, W. E., Waters, C. Z., Denneau, L., Draper, P., Hodapp, K. W., Jedicke, R., Kudritzki, R.-P., Metcalfe, N., Stubbs, C. W., & Wainscoat, R. J. 2016a, ArXiv e-prints
- Magnier, E. A., Sweeney, W. E., Chambers, K. C., Flewelling, H. A., Huber, M. E., Price, P. A., Waters, C. Z., Denneau, L., Draper, P., Jedicke, R., Hodapp, K. W., Kudritzki, R.-P., Metcalfe, N., Stubbs, C. W., & Wainscoat, R. J. 2016b, ArXiv e-prints
- Tonry, J. L., Stubbs, C. W., Lykke, K. R., Doherty, P., Shivvers, I. S., Burgett, W. S., Chambers, K. C., Hodapp, K. W., Kaiser, N., Kudritzki, R.-P., Magnier, E. A., Morgan, J. S., Price, P. A., & Wainscoat, R. J. 2012, ApJ, 750, 99
- Waters, C. Z., Magnier, E. A., Price, P. A., Chambers, K. C., Draper, P., Flewelling, H. A., Hodapp, K. W., Huber, M. E., Jedicke, R., Kaiser, N., Kudritzki, R.-P., Lupton, R. H., Metcalfe, N., Rest, A., Sweeney, W. E., Tonry, J. L., Wainscoat, R. J., Wood-Vasey, W. M., & Builders, P. 2016, ArXiv e-prints